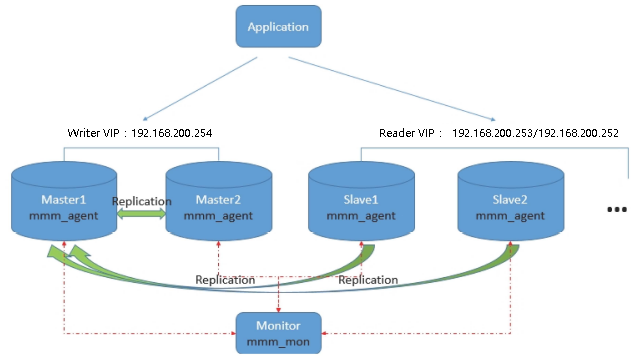
**基于MMM实现MairaDB高可用**

# 一、MMM介绍

MMM（Master-Master Replication Manager for MySQL）是一套支持双主故障切换和双主日常管理的脚本程序。MMM使用Perl语言开发，主要用来监控和管理MySQL Master-Master（双主）复制，虽然叫双主复制，但是业务上同一时刻只允许对一个主进行写入，另一台备选主提供部分读业务。以加速在主主切换时刻备选主的预热，可以说MMM这套脚本程序一方面实现了故障切换的功能，另一方面其内部附加的工具脚本也可以实现多个slave的read负载均衡。

MMM提供了自动和手动两种方式移除一组服务器中复制延迟较高的服务器的虚拟IP，同时它还可以备份数据，实现两节点之间的数据同步等。由于MMM无法完全的保证数据一致性，所以MMM适用于对数据一致性要求不是很高，但是又想最大程序的保证数据业务可用性的场景。对于那些对数据的一致性要求很高的业务，非常不建议采用MMM这种高可用架构。



## MySQL-MMM工作原理

MMM（Master-Master Replication Manager for MySQL，MySQL主主复制管理器）是一套灵活的脚本程序。基于Perl实现，用来对MySQL Replication进行监控和故障迁移，并能管理MySQL Master-Master复制的配置（同一时间只有一个节点是可写的）。

mmm\_mond：监控进程，负责所有的监控工作，决定和处理所有节点角色活动。因此脚本需要在监管机上运行。

mmm\_agentd：运行在每个mysql服务器上的代理进程，完成监控的探针工作和执行简单的远端服务设置。此脚本需要在被监管机上运行。

mmm\_control：一个简单的脚本，提供管理mmm\_mond进程的命令。

mysql-mmm的监管端会提供多个虚拟IP（VIP），包括一个写VIP，多个可读VIP，通过监管的管理，这些IP会绑定在可用MySQL之上，当某台MySQL宕机时，监管会将VIP地址迁移至其他MySQL服务器。

在整个监管过程中，需要在mysql中添加相关授权用户，以便让mysql可以支持监管机的维护，授权的用户包括mmm\_monito用户和一个mmm\_agent用户，如果想使用mmm的备份工具则还要添加一个mmm\_tools用户。

# 二、MMM实现MySQL高可用案例

所有操作系统均为centos 7.x 64bit，涉及到主从复制环境搭建后面会简单演示步骤，但是相关的安全复制不会详细说明、服务器需能连接Internet。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 主机 | IP地址 | 主机名 | server-id | VIP地址 |
| Master1 | 192.168.200.111 | mysql-m1 | 1 | 192.168.200.254 |
| Master2 | 192.168.200.112 | mysql-m2 | 2 | - |
| Slave1 | 192.168.200.113 | mysql-m3 | 3 | 192.168.200.253 |
| Slave2 | 192.168.200.114 | mysql-m4 | 4 | 192.168.200.252 |
| Monitor | 192.168.200.115 | mysql-monitor | - | - |

安装前准备：配置MySQL-M[1-4]

**[root@server01 ~]# echo mysql-m1 >/etc/hostname**

**[root@server01 ~]# hostname mysql-m1**

**[root@server01 ~]# bash**

**[root@mysql-m1 ~]# iptables -F**

**[root@mysql-m1 ~]# systemctl stop firewalld**

**[root@mysql-m1 ~]# setenforce 0**

**setenforce: SELinux is disabled**

2.1、安装MySQL并配置MySQL-M[1-4]

**[root@mysql-m1 ~]# rm -rf /etc/yum.repos.d/\***

**[root@mysql-m1 ~]# cat <<END>/etc/yum.repos.d/y.repo**

**[y]**

**name=y**

**baseurl=file:///media**

**gpgcheck=0**

**END**

**[root@mysql-m1 ~]# mount /dev/cdrom /media/**

**mount: /dev/sr0 写保护，将以只读方式挂载**

**[root@mysql-m1 ~]# yum -y install mariadb mariadb-devel mariadb-server**

**[root@mysql-m1 ~]# systemctl start mariadb**

**[root@mysql-m1 ~]# mysqladmin -u root password '123456'**

**[root@mysql-m1 ~]# netstat -anpt|grep mysql**

**tcp 0 0 0.0.0.0:3306 0.0.0.0:\* LISTEN 3813/mysqld**

2.2、配置MySQL-M[1-2]的主配置文件

**[root@mysql-m1 ~]# vim /etc/my.cnf**

**[mysqld]**

**log-bin=mysql-bin**

**log-slave-updates=true**

**server-id=1 #另一台master设置为2**

**relay-log=relay-log-bin**

**relay-log-index=relay-log-bin.index**

**[root@mysql-m1 ~]# systemctl restart mariadb**

配置MySQL-M[3-4]的主配置文件

**[root@mysql-m1 ~]# vim /etc/my.cnf**

**[mysqld]**

**server-id=3 #另一台slave设置为4**

**relay-log=relay-log-bin**

**relay-log-index=relay-log-bin.index**

**[root@mysql-m1 ~]# systemctl restart mariadb**

2.3、配置MySQL-M1、MySQL-M2的主主复制模式

MySQL-M1、MySQL-M2先查看binlog日志和pos值位置

**[root@mysql-m1 ~]# mysql -uroot -p123456**

**Welcome to the MariaDB monitor. Commands end with ; or \g.**

**Your MariaDB connection id is 2**

**Server version: 5.5.56-MariaDB MariaDB Server**

**Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.**

**Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.**

**MariaDB [(none)]> show master status;**

**+------------------+----------+--------------+------------------+**

**| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |**

**+------------------+----------+--------------+------------------+**

**| mysql-bin.000001| 245| | |**

**+------------------+----------+--------------+------------------+**

**1 row in set (0.00 sec)**

**[root@mysql-m2 ~]# mysql -uroot -p123456**

**Welcome to the MariaDB monitor. Commands end with ; or \g.**

**Your MariaDB connection id is 2**

**Server version: 5.5.56-MariaDB MariaDB Server**

**Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.**

**Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.**

**MariaDB [(none)]> show master status;**

**+------------------+----------+--------------+------------------+**

**| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |**

**+------------------+----------+--------------+------------------+**

**| mysql-bin.000001 | 245 | | |**

**+------------------+----------+--------------+------------------+**

**1 row in set (0.00 sec)**

MySQL-M1、MySQL-M2相互设置访问权限

**[root@mysql-m1 ~]# mysql -uroot -p123456**

**MariaDB [(none)]> grant replication slave on \*.\* to 'replication'@'192.168.200.112' identified by '123456';**

**Query OK, 0 rows affected (0.01 sec)**

**MariaDB [(none)]> flush privileges;**

**Query OK, 0 rows affected (0.01 sec)**

**MariaDB [(none)]> stop slave;**

**Query OK, 0 rows affected, 1 warning (0.01 sec)**

**MariaDB [(none)]> change master to master\_host='192.168.200.112',master\_user='replication',master\_password='123456',master\_log\_file='mysql-bin.000001',master\_log\_pos=245;**

**Query OK, 0 rows affected (0.08 sec)**

**MariaDB [(none)]> start slave;**

**Query OK, 0 rows affected (0.01 sec)**

**[root@mysql-m2 ~]# mysql -uroot -p123456**

**MariaDB [(none)]> grant replication slave on \*.\* to 'replication'@'192.168.200.111' identified by '123456';**

**Query OK, 0 rows affected (0.01 sec)**

**MariaDB [(none)]> flush privileges;**

**Query OK, 0 rows affected (0.01 sec)**

**MariaDB [(none)]> stop slave;**

**Query OK, 0 rows affected, 1 warning (0.00 sec)**

**MariaDB [(none)]> change master to master\_host='192.168.200.111',master\_user='replication',master\_password='123456',master\_log\_file='mysql-bin.000001',master\_log\_pos=245;**

**Query OK, 0 rows affected (0.09 sec)**

**MariaDB [(none)]> start slave;**

**Query OK, 0 rows affected (0.03 sec)**

MySQL-M1、MySQL-M2分别查看服务器主从状态

**[root@mysql-m1 ~]# mysql -uroot -p123456**

**MariaDB [(none)]> show slave status\G**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Slave\_IO\_State: Waiting for master to send event**

**Master\_Host: 192.168.200.112**

**Master\_User: replication**

**Master\_Port: 3306**

**Connect\_Retry: 60**

**Master\_Log\_File: mysql-bin.000003**

**Read\_Master\_Log\_Pos: 621**

**Relay\_Log\_File: relay-log-bin.000002**

**Relay\_Log\_Pos: 767**

**Relay\_Master\_Log\_File: mysql-bin.000003**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

**[root@mysql-m2 ~]# mysql -uroot -p123456**

**MariaDB [(none)]> show slave status\G**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Slave\_IO\_State: Waiting for master to send event**

**Master\_Host: 192.168.200.111**

**Master\_User: replication**

**Master\_Port: 3306**

**Connect\_Retry: 60**

**Master\_Log\_File: mysql-bin.000003**

**Read\_Master\_Log\_Pos: 859**

**Relay\_Log\_File: relay-log-bin.000002**

**Relay\_Log\_Pos: 767**

**Relay\_Master\_Log\_File: mysql-bin.000003**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

两台机器都查看同步状态Slave\_IO和Slave\_SQL都是YES表示主主同步配置成功

测试主主同步在MySQL-M2上新建立一个数据库m2、MySQL-M1上新建立一个数据库m1

**[root@mysql-m1 ~]# mysql -uroot -p123456**

**MariaDB [(none)]> create database m1;**

**Query OK, 1 row affected (0.00 sec)**

**[root@mysql-m2 ~]# mysql -uroot -p123456**

**MariaDB [(none)]> create database m2;**

**Query OK, 1 row affected (0.00 sec)**

**MariaDB [(none)]> show databases;**

**+--------------------+**

**| Database |**

**+--------------------+**

**| information\_schema |**

**| m1 |**

**| m2 |**

**| mysql |**

**| performance\_schema |**

**| test |**

**+--------------------+**

**6 rows in set (0.00 sec)**

回到MySQL-M1服务器上，查看刚才MySQL-M2上建立的数据库信息。

**[root@mysql-m1 ~]# mysql -uroot -p123456**

**MariaDB [(none)]> show databases;**

**+--------------------+**

**| Database |**

**+--------------------+**

**| information\_schema |**

**| m1 |**

**| m2 |**

**| mysql |**

**| performance\_schema |**

**| test |**

**+--------------------+**

**6 rows in set (0.00 sec)**

2.4、配置MySQL-M3、MySQL-M4作为MySQL-M1的从库

先查看MySQL-M1的状态值

**MariaDB [(none)]> show master status;**

**+------------------+----------+--------------+------------------+**

**| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |**

**+------------------+----------+--------------+------------------+**

**| mysql-bin.000001 | 879 | | |**

**+------------------+----------+--------------+------------------+**

**1 row in set (0.00 sec)**

**MariaDB [(none)]> grant replication slave on \*.\* to 'replication'@'192.168.200.113' identified by '123456';**

**Query OK, 0 rows affected (0.00 sec)**

**MariaDB [(none)]> grant replication slave on \*.\* to 'replication'@'192.168.200.114' identified by '123456';**

**Query OK, 0 rows affected (0.01 sec)**

**MariaDB [(none)]> flush privileges;**

**Query OK, 0 rows affected (0.01 sec)**

**MariaDB [(none)]> create database s1;**

**Query OK, 1 row affected (0.00 sec)**

在MySQL-M3、MySQL-M4分别指向M1并查看服务器主从状态

**MariaDB [(none)]> stop slave;**

**Query OK, 0 rows affected, 1 warning (0.00 sec)**

**MariaDB [(none)]> change master to master\_host='192.168.200.111',master\_user='replication',master\_password='123456',master\_log\_file='mysql-bin.000001',master\_log\_pos=879;**

**Query OK, 0 rows affected (0.04 sec)**

**MariaDB [(none)]> start slave;**

**Query OK, 0 rows affected (0.01 sec)**

**MariaDB [(none)]> show slave status\G**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Slave\_IO\_State: Waiting for master to send event**

**Master\_Host: 192.168.200.111**

**Master\_User: replication**

**Master\_Port: 3306**

**Connect\_Retry: 60**

**Master\_Log\_File: mysql-bin.000003**

**Read\_Master\_Log\_Pos: 1497**

**Relay\_Log\_File: relay-log-bin.000002**

**Relay\_Log\_Pos: 1009**

**Relay\_Master\_Log\_File: mysql-bin.000003**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

**MariaDB [(none)]> show databases;**

**+--------------------+**

**| Database |**

**+--------------------+**

**| information\_schema |**

**| mysql |**

**| performance\_schema |**

**| s1 |**

**| test |**

**+--------------------+**

**5 rows in set (0.01 sec)**

2.5、安装mysql-mmm软件包（所有主机安装）

**已经提前下载好所需软件包，在存储目录中执行yum安装即可**

**[root@mysql-m1 ~]# yum -y localinstall \*.rpm**

在mysql-m[1-4]服务器上添加授权使monitor可以访问

**MariaDB [(none)]> grant all on \*.\* to 'mmm\_agent'@'192.168.200.%' identified by '123456';**

**MariaDB [(none)]> flush privileges;**

修改/etc/mysql-mmm/mmm\_common.conf配置文件，系统中所有主机的该配置文件内容都是一样的，包括监控主机mysql-monitor。

**[root@mysql-m1 ~]# vim /etc/mysql-mmm/mmm\_common.conf**

**active\_master\_role writer**

**<host default>**

**cluster\_interface ens33**

**pid\_path /run/mysql-mmm-agent.pid**

**bin\_path /usr/libexec/mysql-mmm/**

**replication\_user replication**

**replication\_password 123456**

**agent\_user mmm\_agent**

**agent\_password 123456**

**</host>**

**<host db1>**

**ip 192.168.200.111**

**mode master**

**peer db2**

**</host>**

**<host db2>**

**ip 192.168.200.112**

**mode master**

**peer db1**

**</host>**

**<host db3>**

**ip 192.168.200.113**

**mode slave**

**</host>**

**<host db4>**

**ip 192.168.200.114**

**mode slave**

**</host>**

**<role writer>**

**hosts db1, db2**

**ips 192.168.200.254**

**mode exclusive** #排它模式

**</role>**

**<role reader>**

**hosts db3, db4**

**ips 192.168.200.253, 192.168.200.252**

**mode balanced #平衡模式**

**</role>**

**[root@mysql-m1 ~]# scp /etc/mysql-mmm/mmm\_common.conf 192.168.200.112:/etc/mysql-mmm/**

**[root@mysql-m1 ~]# scp /etc/mysql-mmm/mmm\_common.conf 192.168.200.113:/etc/mysql-mmm/**

**[root@mysql-m1 ~]# scp /etc/mysql-mmm/mmm\_common.conf 192.168.200.114:/etc/mysql-mmm/**

**[root@mysql-m1 ~]# scp /etc/mysql-mmm/mmm\_common.conf 192.168.200.115:/etc/mysql-mmm/**

在所有数据库主机上我们需要编辑/etc/mysql-mmm/mmm\_agent.conf文件，根据不同的主机修改为不同的值：

**[root@mysql-m1 ~]# vim /etc/mysql-mmm/mmm\_agent.conf**

**include mmm\_common.conf**

**this db1 #分别改为db1、db2、db3、db4**

在监控主机（115）上编辑/etc/mysql-mmm/mmm\_mon.conf配置文件

**[root@mysql-monitor ~]# vim /etc/mysql-mmm/mmm\_mon.conf**

**include mmm\_common.conf**

**<monitor>**

**ip 127.0.0.1**

**pid\_path /run/mysql-mmm-monitor.pid**

**bin\_path /usr/libexec/mysql-mmm**

**status\_path /var/lib/mysql-mmm/mmm\_mond.status**

**ping\_ips 192.168.200.111,192.168.200.112,192.168.200.113,192.168.200.114**

**auto\_set\_online 60**

**# The kill\_host\_bin does not exist by default, though the monitor will**

**# throw a warning about it missing. See the section 5.10 "Kill Host**

**# Functionality" in the PDF documentation.**

**#**

**# kill\_host\_bin /usr/libexec/mysql-mmm/monitor/kill\_host**

**#**

**</monitor>**

**<host default>**

**monitor\_user mmm\_agent**

**monitor\_password 123456**

**</host>**

**debug 0**

启动mysql agent代理服务

**[root@mysql-m1 ~]# systemctl start mysql-mmm-agent**

**[root@mysql-m2 ~]# systemctl start mysql-mmm-agent**

**[root@mysql-m3 ~]# systemctl start mysql-mmm-agent**

**[root@mysql-m4 ~]# systemctl start mysql-mmm-agent**

[root@mysql-m1 ~]# systemctl start mysql-mmm-agent

[root@mysql-m1 ~]# netstat -anpt|grep mmm

tcp 0 0 192.168.200.111:9989 0.0.0.0:\* LISTEN 13071/mmm\_agentd

启动监控

**[root@mysql-monitor ~]# systemctl start mysql-mmm-monitor**

[root@mysql-monitor src]# netstat -anpt|grep mmm

tcp 0 0 127.0.0.1:9988 0.0.0.0:\* LISTEN 12777/mmm\_mond

测试集群

**[root@mysql-monitor ~]# mmm\_control show**

**db1(192.168.200.111) master/ONLINE. Roles: writer(192.168.200.254)**

**db2(192.168.200.112) master/ONLINE. Roles:**

**db3(192.168.200.113) slave/ONLINE. Roles: reader(192.168.200.252)**

**db4(192.168.200.114) slave/ONLINE. Roles: reader(192.168.200.253)**

由此看来，主mysql-m1是对外提供一个写入的角色，可通过amoeba等软件结合，实现读写分离，后面的虚拟IP是真正来访问mysql数据库的地址。

故障转移切换

停掉主db1数据库，等待几秒后，可以看到数据库db1处于HARD\_offline(离线状态)，检测不到数据库的存在。DOWN掉mysql-m1数据库，虚拟IP地址会全部在另一台正常数据库上。

**[root@mysql-m1 ~]# systemctl stop mariadb**

**[root@mysql-monitor ~]# mmm\_control show**

**db1(192.168.200.111) master/HARD\_OFFLINE. Roles:**

**db2(192.168.200.112) master/ONLINE. Roles: writer(192.168.200.254)**

**db3(192.168.200.113) slave/ONLINE. Roles: reader(192.168.200.253)**

**db4(192.168.200.114) slave/ONLINE. Roles: reader(192.168.200.252)**

**[root@mysql-m4 ~]# mysql -p123456 #查看主从复制中主服务器已经切换为112了**

**MariaDB [(none)]> show slave status\G;**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Slave\_IO\_State: Waiting for master to send event**

**Master\_Host: 192.168.200.112**

**Master\_User: replication**

**Master\_Port: 3306**

**Connect\_Retry: 60**

**Master\_Log\_File: mysql-bin.000001**

**Read\_Master\_Log\_Pos: 2897**

**Relay\_Log\_File: relay-log-bin.000002**

**Relay\_Log\_Pos: 529**

**Relay\_Master\_Log\_File: mysql-bin.000001**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

**Replicate\_Do\_DB:**

**Replicate\_Ignore\_DB:**

**Replicate\_Do\_Table:**

**Replicate\_Ignore\_Table:**

**Replicate\_Wild\_Do\_Table:**

**Replicate\_Wild\_Ignore\_Table:**

**Last\_Errno: 0**

**Last\_Error:**

**Skip\_Counter: 0**

**Exec\_Master\_Log\_Pos: 2897**

**Relay\_Log\_Space: 821**

**Until\_Condition: None**

**Until\_Log\_File:**

**Until\_Log\_Pos: 0**

**Master\_SSL\_Allowed: No**

**Master\_SSL\_CA\_File:**

**Master\_SSL\_CA\_Path:**

**Master\_SSL\_Cert:**

**Master\_SSL\_Cipher:**

**Master\_SSL\_Key:**

**Seconds\_Behind\_Master: 0**

**Master\_SSL\_Verify\_Server\_Cert: No**

**Last\_IO\_Errno: 0**

**Last\_IO\_Error:**

**Last\_SQL\_Errno: 0**

**Last\_SQL\_Error:**

**Replicate\_Ignore\_Server\_Ids:**

**Master\_Server\_Id: 2**

**1 row in set (0.00 sec)**

**ERROR: No query specified**

启动主db1数据库后，可以看到数据库db1处于AWAITING\_RECOVERY(恢复状态)，几秒后将恢复到在线状态。但不会抢占112的主库角色，如更改需要手动切换

**[root@mysql-m1 ~]# systemctl start mariadb**

**[root@mysql-monitor ~]# mmm\_control show**

**db1(192.168.200.111) master/AWAITING\_RECOVERY. Roles:**

**db2(192.168.200.112) master/ONLINE. Roles: writer(192.168.200.254)**

**db3(192.168.200.113) slave/ONLINE. Roles: reader(192.168.200.253)**

**db4(192.168.200.114) slave/ONLINE. Roles: reader(192.168.200.252)**

**[root@mysql-monitor ~]# mmm\_control show**

**db1(192.168.200.111) master/ONLINE. Roles:**

**db2(192.168.200.112) master/ONLINE. Roles: writer(192.168.200.254)**

**db3(192.168.200.113) slave/ONLINE. Roles: reader(192.168.200.253)**

**db4(192.168.200.114) slave/ONLINE. Roles: reader(192.168.200.252)**

**[root@mysql-monitor ~]# mmm\_control checks all #检查所有数据库的状态**

**db4 ping [last change: 2019/04/29 14:07:16] OK**

**db4 mysql [last change: 2019/04/29 14:07:16] OK**

**db4 rep\_threads [last change: 2019/04/29 14:16:41] OK**

**db4 rep\_backlog [last change: 2019/04/29 14:07:16] OK: Backlog is null**

**db2 ping [last change: 2019/04/29 14:07:16] OK**

**db2 mysql [last change: 2019/04/29 14:07:16] OK**

**db2 rep\_threads [last change: 2019/04/29 14:07:16] OK**

**db2 rep\_backlog [last change: 2019/04/29 14:07:16] OK: Backlog is null**

**db3 ping [last change: 2019/04/29 14:07:16] OK**

**db3 mysql [last change: 2019/04/29 14:07:16] OK**

**db3 rep\_threads [last change: 2019/04/29 14:07:16] OK**

**db3 rep\_backlog [last change: 2019/04/29 14:07:16] OK: Backlog is null**

**db1 ping [last change: 2019/04/29 14:07:16] OK**

**db1 mysql [last change: 2019/04/29 14:21:46] OK**

**db1 rep\_threads [last change: 2019/04/29 14:07:16] OK**

**db1 rep\_backlog [last change: 2019/04/29 14:07:16] OK: Backlog is null**

手动切换master角色

**[root@localhost ~]# mmm\_control move\_role writer db1**

**OK: Role 'writer' has been moved from 'db2' to 'db1'. Now you can wait some time and check new roles info!**

**再次查看主数据库已经为111了，而且所有从库也重新设置主库为111了**

**[root@mysql-monitor ~]# mmm\_control show**

**db1(192.168.200.111) master/ONLINE. Roles: writer(192.168.200.254)**

**db2(192.168.200.112) master/ONLINE. Roles:**

**db3(192.168.200.113) slave/ONLINE. Roles: reader(192.168.200.253)**

**db4(192.168.200.114) slave/ONLINE. Roles: reader(192.168.200.252)**

**[root@mysql-m4 ~]# mysql -p123456**

**MariaDB [(none)]> show slave status\G;**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Slave\_IO\_State: Waiting for master to send event**

**Master\_Host: 192.168.200.111**

**Master\_User: replication**

**Master\_Port: 3306**

**Connect\_Retry: 60**

**Master\_Log\_File: mysql-bin.000002**

**Read\_Master\_Log\_Pos: 245**

**Relay\_Log\_File: relay-log-bin.000002**

**Relay\_Log\_Pos: 529**

**Relay\_Master\_Log\_File: mysql-bin.000002**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

**Replicate\_Do\_DB:**

**Replicate\_Ignore\_DB:**

**Replicate\_Do\_Table:**

**Replicate\_Ignore\_Table:**

**Replicate\_Wild\_Do\_Table:**

**Replicate\_Wild\_Ignore\_Table:**

**Last\_Errno: 0**

**Last\_Error:**

**Skip\_Counter: 0**

**Exec\_Master\_Log\_Pos: 245**

**Relay\_Log\_Space: 821**

**Until\_Condition: None**

**Until\_Log\_File:**

**Until\_Log\_Pos: 0**

**Master\_SSL\_Allowed: No**

**Master\_SSL\_CA\_File:**

**Master\_SSL\_CA\_Path:**

**Master\_SSL\_Cert:**

**Master\_SSL\_Cipher:**

**Master\_SSL\_Key:**

**Seconds\_Behind\_Master: 0**

**Master\_SSL\_Verify\_Server\_Cert: No**

**Last\_IO\_Errno: 0**

**Last\_IO\_Error:**

**Last\_SQL\_Errno: 0**

**Last\_SQL\_Error:**

**Replicate\_Ignore\_Server\_Ids:**

**Master\_Server\_Id: 1**

手动切换在线状态

[root@mysql-monitor src]# mmm\_control set\_online|set\_offline db2

OK: State of 'db2' changed to ONLINE. Now you can wait some time and check its new roles!